National Institute of Technology, Warangal

Dept. of Computer Science and Engineering

VR Shooting Game

Presented by:

Upre Ashish Kumar 21CSB0B62

Abhinash Kumar Shah 21CSB0F37

Presented to: Dr. Ijjina Earnest Paul

Outline

1. Introduction
2. Literature Review
3. Methodology
    - Development Process
    - System Architecture and Workflow
4. Implementation
5. Testing & Evaluation
6. Results & Discussion
7. Conclusion
8. References

Introduction

- **VR technology:**
  - Very influential in gaming.
  - High level of immersion useful for game experience.
  - Unity is a cross-platform game engine used to create 2D and 3D interactive experiences, including games, simulations, and other applications.
  - In this project we have made use of Unity for the development of a shooting game.

Problem Statement:

- Traditional shooting games lack immersivity.
- This project aims to createa a VR shooting game that gives a compelling experience.
- Aims to enhance gaming experience through an interactive VR environment.

Introduction

Objectives and scope:

- Develop a compelling VR-based shooting game.
- Implement ray gun mechanics for shooting bullets.
- Implement enemy spawning.
- Use Unity XR Device Simulator of XR Interaction toolkit for simulation.

Significance:

- Demonstrates potential of VR.
- Hands-on learning of VR development

Literature Review (1/2)

- VR technology creates a simulated environment using specialized hardware like VR headsets and motion controllers.
- AR overlays virtual objects on the real world, while VR offers fully immersive experiences.

Exising VR applications for gaming:

- Beat Saber: A VR rhythm game with dynamic sword-slashing mechanics.
- Superhot VR: A first-person shooter with time-controlled movement.

Literature Review (2/2)

Feature	Ours	Superhot VR	Arizona Sunshine
Graphics	Simple models and textures	Low-poly, minimalist style	Realistic graphics
Enemy AI	Basic collision-based logic	Time-controlled AI	Advanced zombie behavior
Gameplay	Point-and-shoot interaction	Bullet-dodging mechanics	Survival shooting
Technology	Unity XR Interaction Toolkit	Custom VR engine	Unity Engine
Spawning	Randomized enemy spawning	No spawning (static waves)	Scripted enemy waves

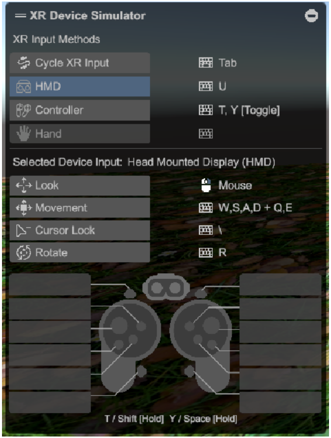
Table: Comparison of VR Shooting Games

Methodology

Tools and Technologies Used:

1. **Game Engine:**
   - Unity 6 was used for the implementation of the project.
2. **VR Simulation:**
   - Since a VR headset was unavailable, we used the XR Interaction Toolkit package in Unity.
   - The XR Device Simulator, a prefab provided by the XR Interaction Toolkit, was used to simulate the VR experience.
   - The XR Device Simulator was attached to the scene, allowing control of the player using on-screen controls.
3. **Programming Language:**
   - C-Sharp (C#) was used for scripting in Unity.
   - Scripts were attached to objects to modify their behavior dynamically.
4. **Development Environment:**
   - Visual Studio 2022 was used as the IDE due to its ease of use and seamless integration with Unity 6.

Methodology



XR Device Simulator

XR Input Methods

Cycle XR Input	Tab
HMD	U
Controller	T, Y [Toggle]
Hand	

Selected Device Input: Head Mounted Display (HMD)

Look	Mouse
Movement	W,S,A,D + Q,E
Cursor Lock	\
Rotate	R

T / Shift [Hold] Y / Space [Hold]

Figure: Controls of XR Device Simulator

Methodology

Agile Development Approach We have followed an Agile approach for the development of the project. We brainstormed the design and started the development process iteratively. This approach allowed us to accommodate changes without freezing any design choices or requirements. We solved problems encountered during development through debugging and alternative approaches when necessary. This included changes in the choice of technology well into the project's lifecycle.

Hardware Components

- **VR Headset & Controllers** – Used for player interaction (shooting, aiming, movement).
- **PC (Development Machine)** – Runs Unity, handling game logic, physics, and rendering.

Software Components

- **Game Engine (Unity 6)** – Manages rendering, physics, interactions, and animations.
- **XR Interaction Toolkit** – Handles VR controller input and interactions.
- **C# Scripts** – Implements game logic (shooting mechanics, enemy AI, movement, etc.).

Methodology

Functional Modules

Player Controller

- **RayGun.cs** – Handles shooting, bullet instantiation, and input detection.
- **XR Rig (VR Player Setup)** – Controls player movement and camera view.

Enemy System

- **EnemySpawner.cs** – Spawns enemies at random locations.
- **EnemyBehavior.cs** – Controls enemy movement, reaction to bullets, and destruction.

Interaction System

- **VR Input Handling** – Detects trigger presses for shooting.
- **Collision Detection** – Registers when bullets hit enemies.

Game Environment

- **Ground Plane** – Defines the playable area.

Implementation

Virtual Environment

- Designed to resemble a dynamic combat zone.
- Incorporated obstacles and varied terrains for enhanced realism.
- Ensured appropriate scale and spatial arrangement for comfortable navigation.

User Interface (UI)

- Minimalistic UI displaying essential information such as ammunition count and health status.
- Ensured an unobstructed view to maintain player immersion while conveying critical gameplay data.

Implementation

Interaction and Gameplay

- Leveraged **XR Interaction Toolkit** for natural user interactions.
- Players could aim and shoot using motion controllers.
- Integrated haptic feedback to provide tactile responses, enhancing realism.

Features and Functionalities

- **Realistic Shooting Mechanics** – Implemented a ray gun system for precise aiming and firing.
- **Dynamic Enemy Behavior** – Developed AI that reacts to player actions, takes damage, and gets destroyed upon impact.
- **Randomized Enemy Spawning** – Introduced varied enemy spawn locations and intervals for unpredictable encounters.
- **Interactive Environment** – Included objects that players can interact with, increasing immersion.

Implementation - Challenges Faced and Solutions

Development Challenges

- **No VR Headset** – Development proceeded without access to physical VR hardware.
- Initially used **Meta XR All-in-One SDK**, but its simulator displayed a blank white screen.
- Lack of documentation made debugging impossible.
- Identified a **bug in Unity 6**, tested in Unity 2022 & 2023, but the issue persisted.
- Ultimately switched to **XR Interaction Toolkit**, utilizing the XR Device Simulator for testing.

Final Solution

- Successfully implemented VR interactions using **XR Interaction Toolkit**.
- The simulator provided a functional environment for testing without a physical VR headset.

Testing & Evaluation - Testing Methods

Comprehensive Testing Process

- Ensured functionality, performance, and user experience through structured testing.

User Testing

- We conducted hands-on testing as users.
- Evaluated immersivity and responsiveness of the game.

Performance Testing

- Monitored resource utilization during gameplay.
- Ensured smooth performance without lag or stuttering.

Testing & Evaluation - Performance and User Experience Analysis

Performance Analysis

- Maintained a stable frame rate above 60 FPS on VR-capable hardware.
- Implemented efficient memory management, removing bullets and enemies after a set duration.

User Experience Analysis

- **Immersion** – High level of immersion due to responsive controls and realistic enemy interactions.
- **Controls** – XR Interaction Toolkit facilitated natural interactions, but controls felt unintuitive due to lack of a VR headset.
- **Motion Sickness** – Some discomfort experienced during prolonged sessions.

Feedback and Improvements

Key Areas for Improvement

- **Enhanced Visuals** – More detailed models and textures to improve immersion.
- **Advanced Enemy AI** – Implementing strategic movement and varied attack patterns for greater challenge.
- **Expanded Content** – Additional levels, enemy types, and weapons to increase replayability.
- **Motion Sickness Mitigation** – Adjustable movement speeds, teleportation options, and dynamic field-of-view adjustments to reduce discomfort.

Key Findings and Observations

Successful Implementation

- The VR Shooting Game delivers an immersive and interactive experience using Unity Engine and XR Interaction Toolkit.

Key Features and Outcomes

- **Realistic Shooting Mechanics** – Smooth aiming and shooting with a responsive ray gun system.
- **Dynamic Enemy Behavior**
- **Randomized Enemy Spawning** – Unique and unpredictable enemy encounters each session.
- **Good VR Interaction** – XR Interaction Toolkit enables movement and interaction.

Limitations and Future Enhancements

Limitations

- **Basic Graphics**
- **Limited Enemy AI**
- **Static Environment**
- **No Multiplayer Mode**
- **Simplified Bullet Physics** – No realistic bullet drop, ricochets, or advanced interactions.

Future Enhancements

- **Improved Graphics** – High-quality textures, lighting, and detailed models.
- **Advanced Enemy AI** – Implementing pathfinding algorithms (A*, NavMesh).
- **Expanded Gameplay** – More levels and dynamic environments.
- **Multiplayer Mode** – Team-based or PvP gameplay for added engagement.
- **Enhanced Bullet Physics** – Realistic trajectories, impact effects, and particle systems.
- **UI Improvements** – Score counters, health bars, and game timers.
- **VR Haptics and Sound Effects** – Improved sensory feedback for deeper immersion.

Conclusion

Summary of Achievements

- Successfully developed a functional VR shooting game using Unity and XR Interaction Toolkit.
- Implemented realistic ray gun mechanics for smooth and responsive shooting.
- Designed enemy that reacts to collisions, changes color, and gets destroyed after a delay.
- Introduced random enemy spawning for varied gameplay and replayability.

Future Scope

- **Expanded Levels & Complex Terrains** – New environments like forests and deserts for more engaging gameplay.
- **Multiplayer Support** – Enable cooperative and competitive VR gameplay.
- **Enhanced Enemy AI** – Implement smarter AI with diverse attack patterns and behaviors.
- **Power-ups & Scoring System** – Introduce collectibles and a scoring mechanism to enhance player progression.
- **Graphics & Animations** – Improve textures, lighting effects, and animations for better realism.

References

For references, please refer to the project report.

Q&A

Thank you!

We welcome your questions and feedback.